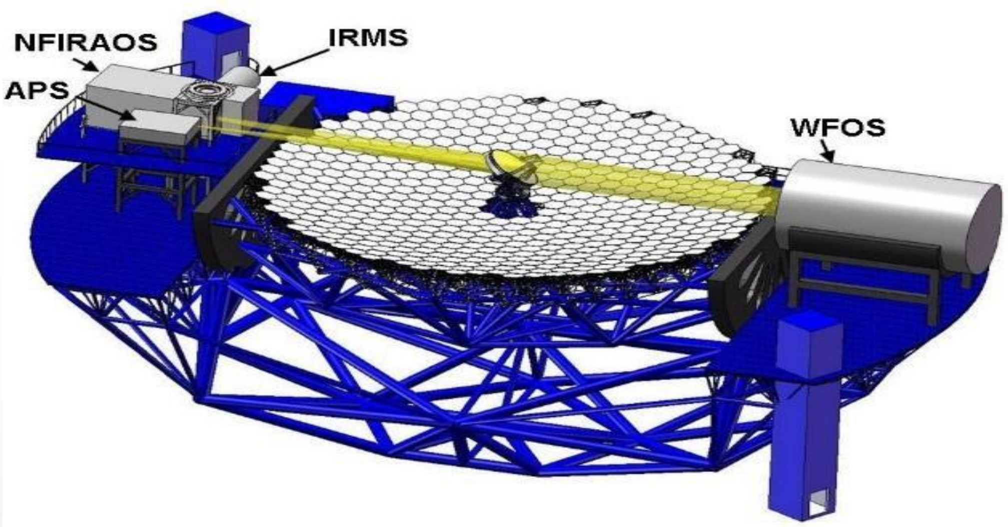
Jet Propulsion Laboratory
California Institute of Technology

Verifying Interface Specifications and Generating ICDs for the APS of the TMT from a System Model in SysML

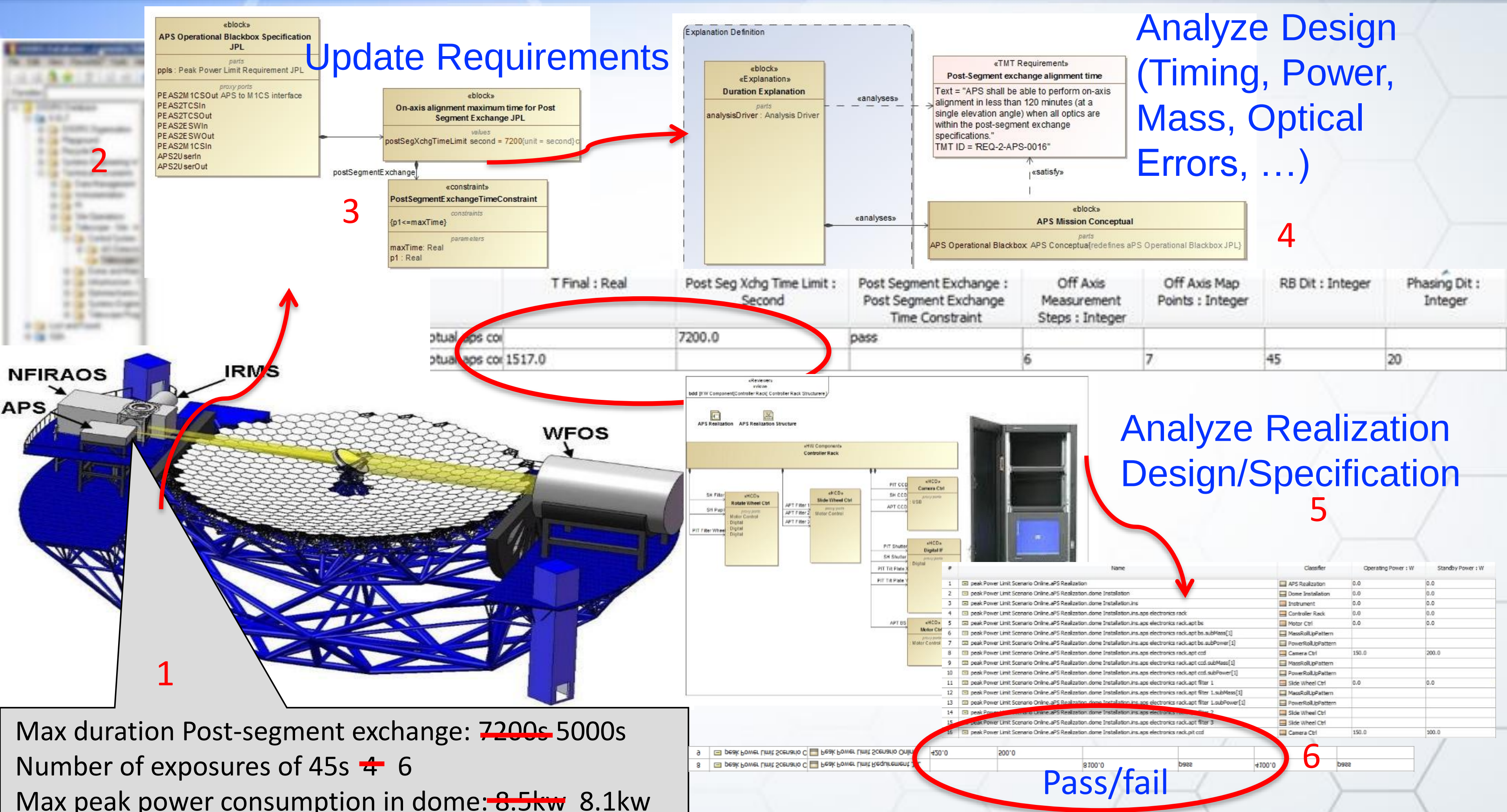
Sebastian Herzig, Robert Karban, Gary Brack, Scott B. Michaels, Frank Dekens, Mitchell Troy

Context

- Alignment and Phasing System (APS)
  - Sensor responsible for measuring the pre-adaptive optics wavefront quality
  - APS (and AO) team uses MBSE with SysML to analyze requirements, produce design, and perform analysis



TMT / APS MBSE Approach


NFIRAOS
APS
IRMS
WFOS
1
2
3
4
5
6
Explanation Definition
«analyses»
«analyses»
«satisfy»
postSegmentExchange
«block» APS Operational Blackbox Specification JPL
parts
ppls : Peak Power Limit Requirement JPL
proxy ports
PEAS2M1CSOut APS to M1CS interface
PEAS2TCSIn
PEAS2TCSOut
PEAS2ESWIn
PEAS2ESWOut
PEAS2M1CSIn
APS2UserIn
APS2UserOut
«block» On-axis alignment maximum time for Post Segment Exchange JPL
values
postSegXchgTimeLimit second = 7200{unit = second}
«constraint» PostSegmentExchangeTimeConstraint
constraints
{p1<=maxTime}
parameters
maxTime: Real
p1 : Real
«block» «Explanation» Duration Explanation
parts
analysisDriver : Analysis Driver
«block» APS Mission Conceptual
parts
APS Operational Blackbox: APS Conceptual{redefines aPS Operational Blackbox JPL}
«HW Component» Controller Rack
«HCD» Rotate Wheel Ctrl
«HCD» Slide Wheel Ctrl
«HCD» Camera Ctrl
«HCD» Digital IF
«HCD» Motor Ctrl
APS Realization
APS Realization Structure
Pass/fail

«TMT Requirement»
Post-Segment exchange alignment time

Text = "APS shall be able to perform on-axis alignment in less than 120 minutes (at a single elevation angle) when all optics are within the post-segment exchange specifications."
TMT ID = 'REQ-2-APS-0016"

	T Final : Real	Post Seg Xchg Time Limit : Second	Post Segment Exchange : Post Segment Exchange Time Constraint	Off Axis Measurement Steps : Integer	Off Axis Map Points : Integer	RB Dit : Integer	Phasing Dit : Integer
ptual aps co		7200.0	pass				
ptual aps co	1517.0			6	7	45	20

#	Name	Classifier	Operating Power : W	Standby Power : W
1	peak Power Limit Scenario Online.aPS Realization	APS Realization	0.0	0.0
2	peak Power Limit Scenario Online.aPS Realization.dome Installation	Dome Installation	0.0	0.0
3	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins	Instrument	0.0	0.0
4	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack	Controller Rack	0.0	0.0
5	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt bs	Motor Ctrl	0.0	0.0
6	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt bs.subMass[1]	MassRollUpPattern		
7	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt bs.subPower[1]	PowerRollUpPattern		
8	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt ccd	Camera Ctrl	150.0	200.0
9	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt ccd.subMass[1]	MassRollUpPattern		
10	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt ccd.subPower[1]	PowerRollUpPattern		
11	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt filter 1	Slide Wheel Ctrl	0.0	0.0
12	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt filter 1.subMass[1]	MassRollUpPattern		
13	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt filter 1.subPower[1]	PowerRollUpPattern		
14	peak Pow[illegible] Online.aPS Realization.dome Installation.ins.aps electronics [illegible] filter 2	Slide Wheel Ctrl		
15	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.apt filter 3	Slide Wheel Ctrl		
16	peak Power Limit Scenario Online.aPS Realization.dome Installation.ins.aps electronics rack.pit ccd	Camera Ctrl	150.0	100.0

Update Requirements

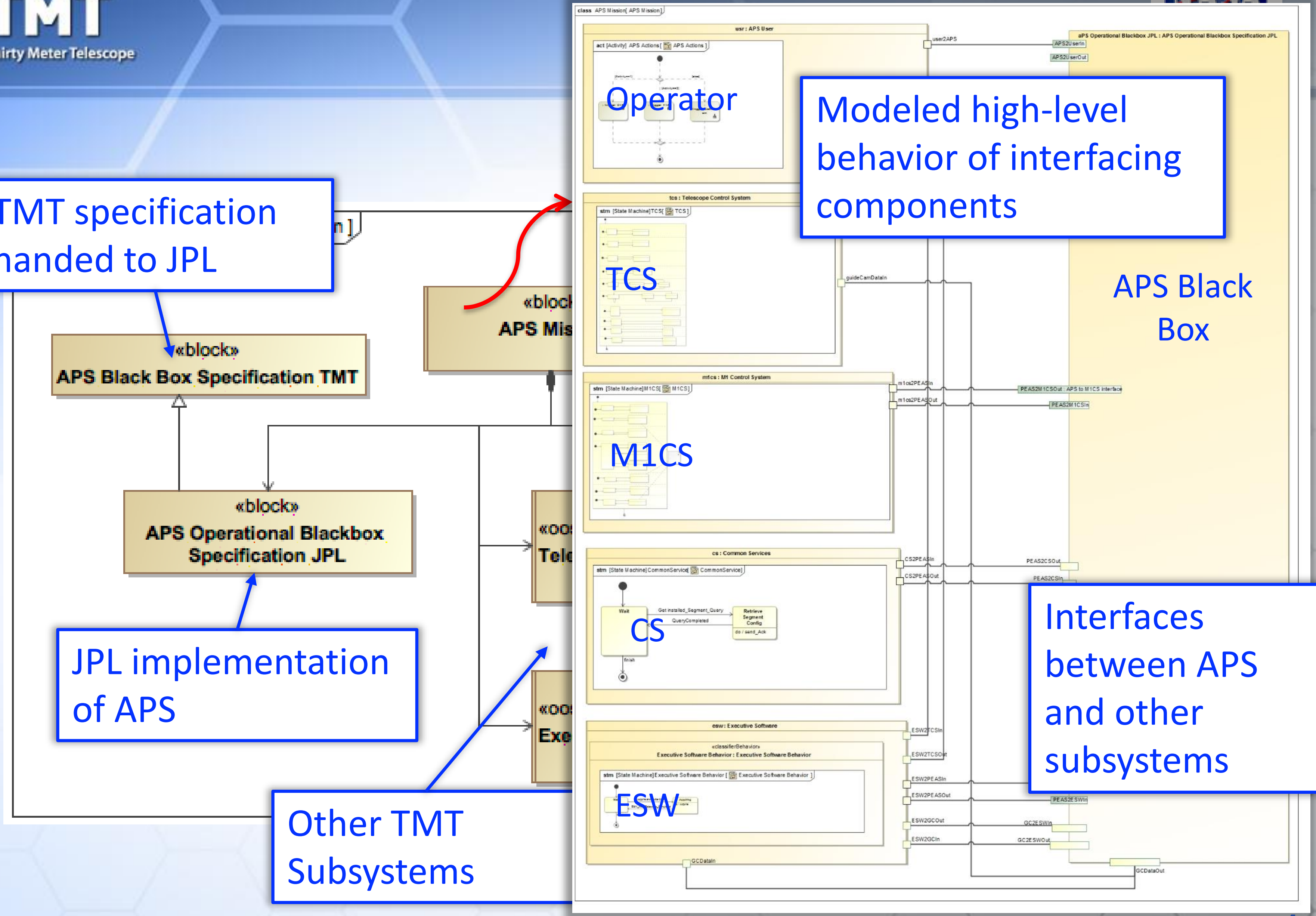
Analyze Design (Timing, Power, Mass, Optical Errors, …)

Analyze Realization Design/Specification

Max duration Post-segment exchange: ~~7200s~~ 5000s
Number of exposures of 45s ~~4~~ 6
Max peak power consumption in dome: ~~8.5kw~~ 8.1kw
Number of motors with 50W ~~10~~ 12

Integrated, Executable System Model

The TMT / APS System Model


«block»
APS Black Box Specification TMT
«block»
APS Operational Blackbox Specification JPL
«block»
APS Mis
«OOS
Tele
«OOS
Exe
class APS Mission[APS Mission]
usr : APS User
act [Activity] APS Actions [APS Actions]
Operator
user2APS
aPS Operational Blackbox JPL : APS Operational Blackbox Specification JPL
APS2UserIn
APS2UserOut
tcs : Telescope Control System
stm [State Machine]TCS[TCS]
TCS
guideCamDataIn
mtcs : M1 Control System
stm [State Machine]M1CS[M1CS]
M1CS
m1cs2PEASIn
m1cs2PEASOut
PEAS2M1CSOut : APS to M1CS interface
PEAS2M1CSIn
APS Black Box
cs : Common Services
stm [State Machine]CommonServices[CommonServices]
CS
Wait
Get installed_Segment_Query
QueryCompleted
Retrieve Segment Config
do / send_Ack
finish
CS2PEASIn
CS2PEASOut
PEAS2CSOut
PEAS2CSIn
esw : Executive Software
«classifierBehavior»
Executive Software Behavior : Executive Software Behavior
stm [State Machine]Executive Software Behavior [Executive Software Behavior]
ESW
ESW2TCSIn
ESW2TCSOut
ESW2PEASIn
ESW2PEASOut
PEAS2ESWIn
ESW2GCOut
ESW2GCIn
GC2ESWIn
GC2ESWOut
GCDataIn
GCDataOut

TMT specification handed to JPL

JPL implementation of APS

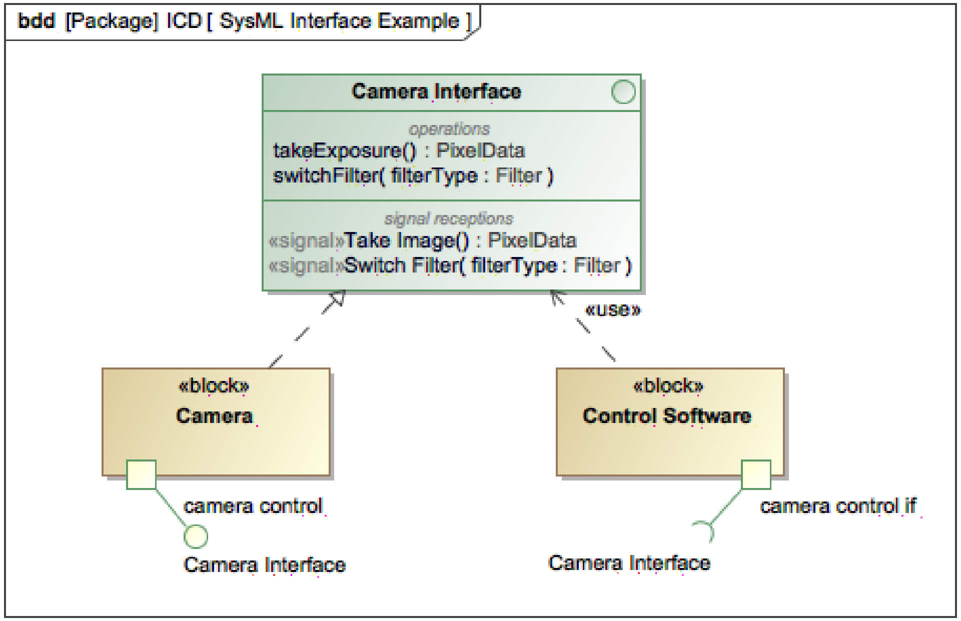
Other TMT Subsystems

Modeled high-level behavior of interfacing components

Interfaces between APS and other subsystems

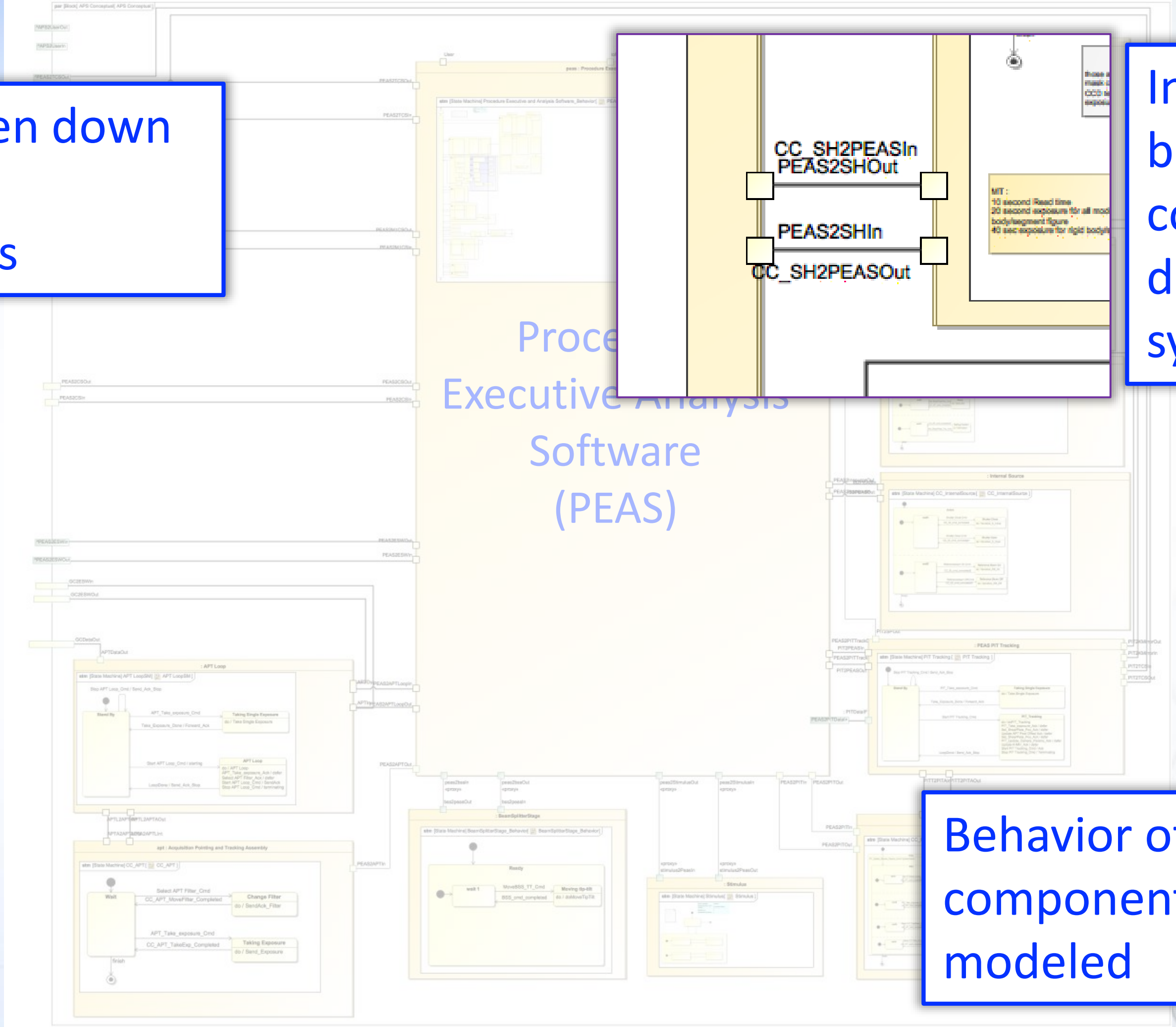
Challenge

- APS interfaces with various other TMT subsystems
- In TMT, interfaces between subsystems are centrally managed in a dedicated system (TMT ICD database)
- Interfaces *can* be modeled in UML / SysML, but no formal link exists between interfaces and behavior in UML / SysML (semantic variation point)
- Idea: static interpretation of subsystem interactions, extract interfaces
- Can we derive interfaces from behavior to verify change controlled interfaces & even generate ICDs? ➔ **Focus on software interfaces**


bdd [Package] ICD [SysML Interface Example]
Camera Interface
operations
takeExposure() : PixelData
switchFilter(filterType : Filter)
signal receptions
«signal»Take Image() : PixelData
«signal»Switch Filter(filterType : Filter)
«use»
«block»
Camera
«block»
Control Software
camera control
Camera Interface
camera control if
Camera Interface

APS / TMT Component Interaction

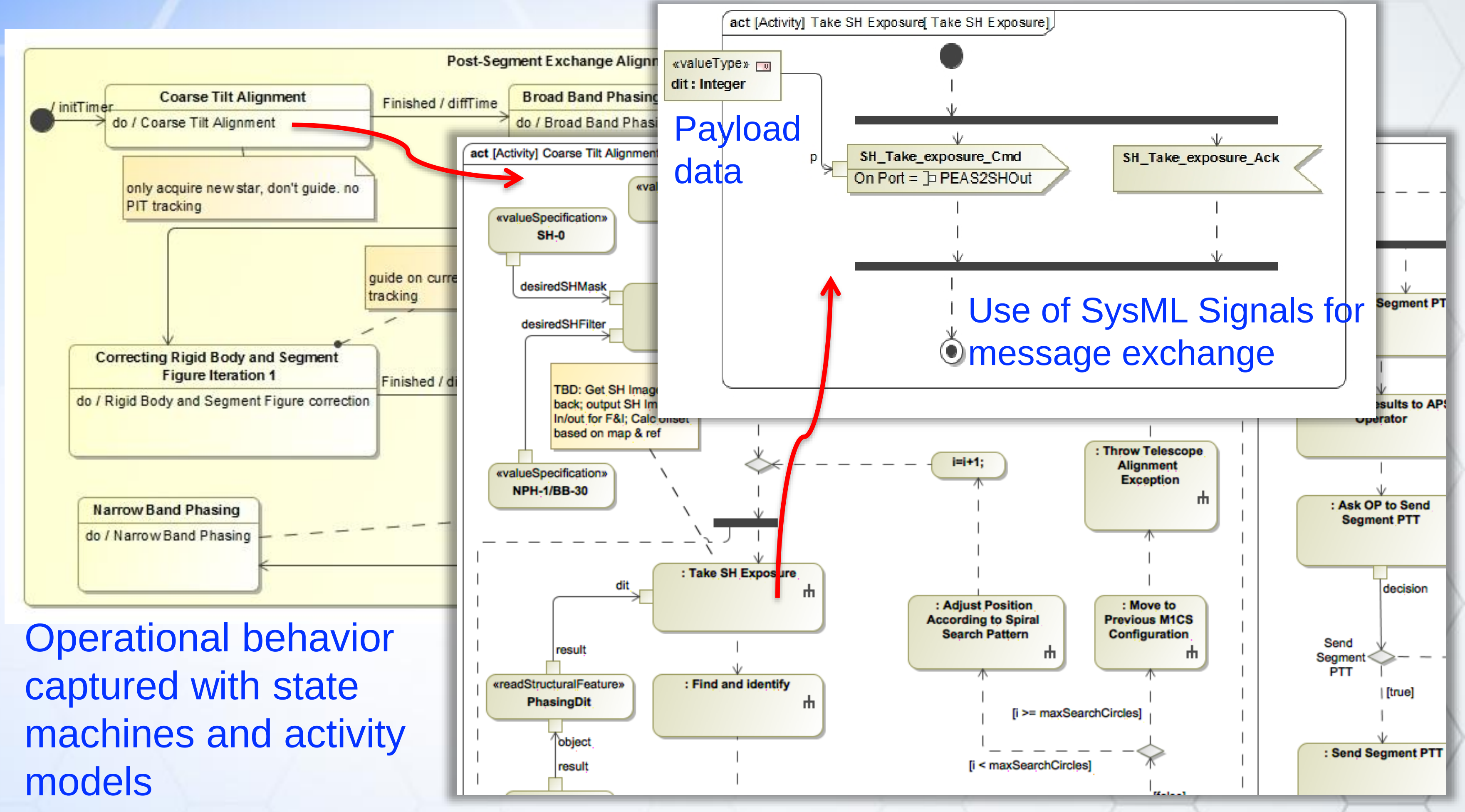
APS is broken down into several components


Procedure Executive Analysis Software (PEAS)
CC_SH2PEASIn
PEAS2SHOut
PEAS2SHIn
CC_SH2PEASOut
MT :
10 second Read time
20 second exposure for all mod
body/segment figure
40 sec exposure for rigid body

Interfaces between components declared in system model

Behavior of *all* components modeled

TMAPS / TMT Component Interactions



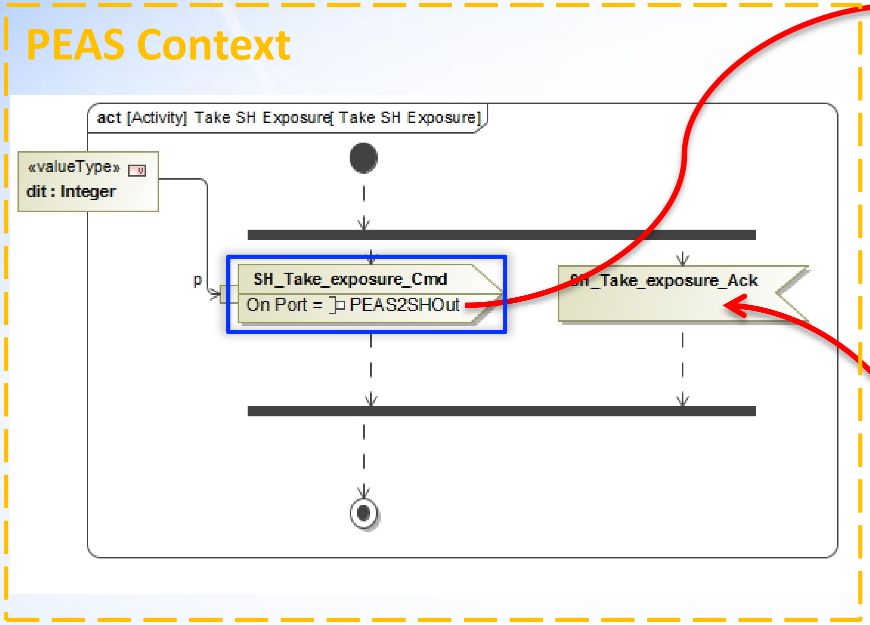
Post-Segment Exchange Alignm
/ initTimer
Coarse Tilt Alignment
do / Coarse Tilt Alignment
Finished / diffTime
Broad Band Phasing
do / Broad Band Phasi
only acquire new star, don't guide. no PIT tracking
guide on curre tracking
Correcting Rigid Body and Segment Figure Iteration 1
do / Rigid Body and Segment Figure correction
Finished / di
Narrow Band Phasing
do / Narrow Band Phasing
act [Activity] Coarse Tilt Alignment
«valueSpecification» SH-0
desiredSHMask
desiredSHFilter
TBD: Get SH Image back; output SH Im In/out for F&I; Calc offset based on map & ref
«valueSpecification» NPH-1/BB-30
dit
: Take SH Exposure
result
«readStructuralFeature» PhasingDit
object
result
: Find and identify
i=i+1;
: Throw Telescope Alignment Exception
: Adjust Position According to Spiral Search Pattern
: Move to Previous M1CS Configuration
[i >= maxSearchCircles]
[i < maxSearchCircles]
Segment PT
esults to APS Operator
: Ask OP to Send Segment PTT
decision
Send Segment PTT
[true]
: Send Segment PTT
act [Activity] Take SH Exposure[Take SH Exposure]
«valueType» dit : Integer
p
SH_Take_exposure_Cmd
On Port = PEAS2SHOut
SH_Take_exposure_Ack

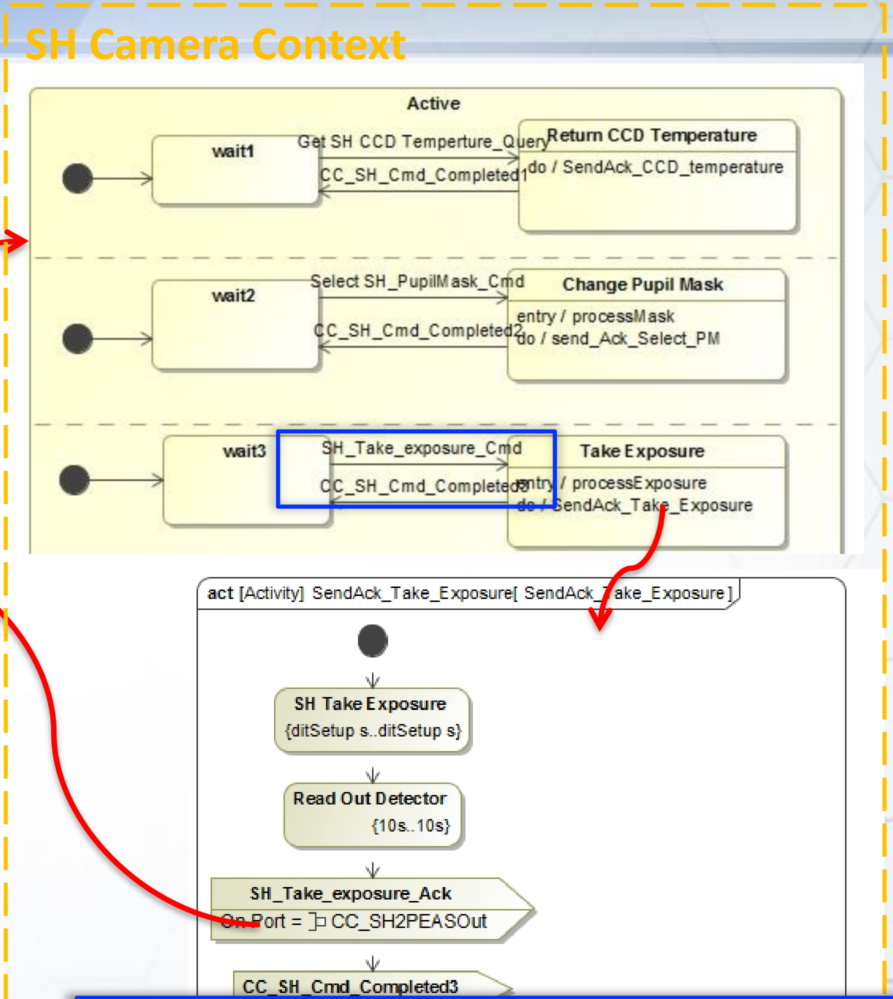
Payload data

Use of SysML Signals for message exchange

Operational behavior captured with state machines and activity models

TMAPS / TMT Component Interactions





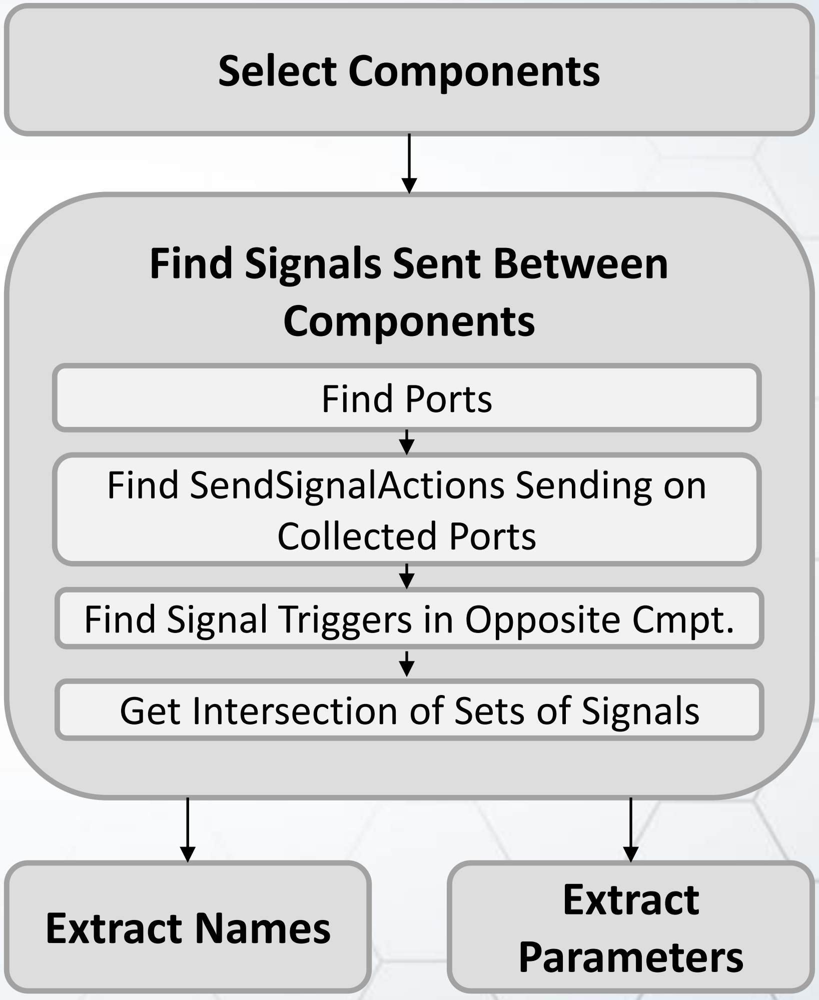
Use of signals sent over ports to simulate a message passing mechanism between components

Same mechanism across subsystems! (e.g., APS to M1CS)

Extracting Software Interfaces

- Focused on extracting software commands, events, parameters
  - Basis: signal exchanges
  - Transformation written in Java
  - Not extracted: publish frequency, timing, error handling, data ranges, protocol details

ICD Concept	SysML Construct
Component	Block
Interface (Declaration)	Port, Connector
Protocol	State Machine, Activity
Command	Signal
Subscribe Event	Signal
Publish Event	Signal
Parameter	Property
Returned Data	Property
Data Type	Value Type, Block



Select Components

Find Signals Sent Between Components

Find Ports

Find SendSignalActions Sending on Collected Ports

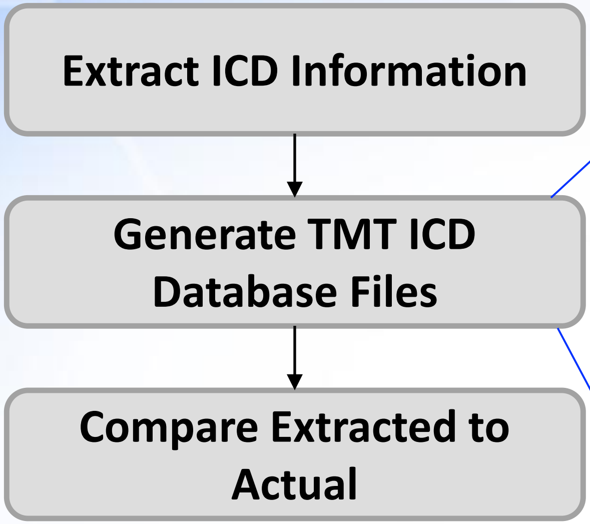
Find Signal Triggers in Opposite Cmpt.

Get Intersection of Sets of Signals

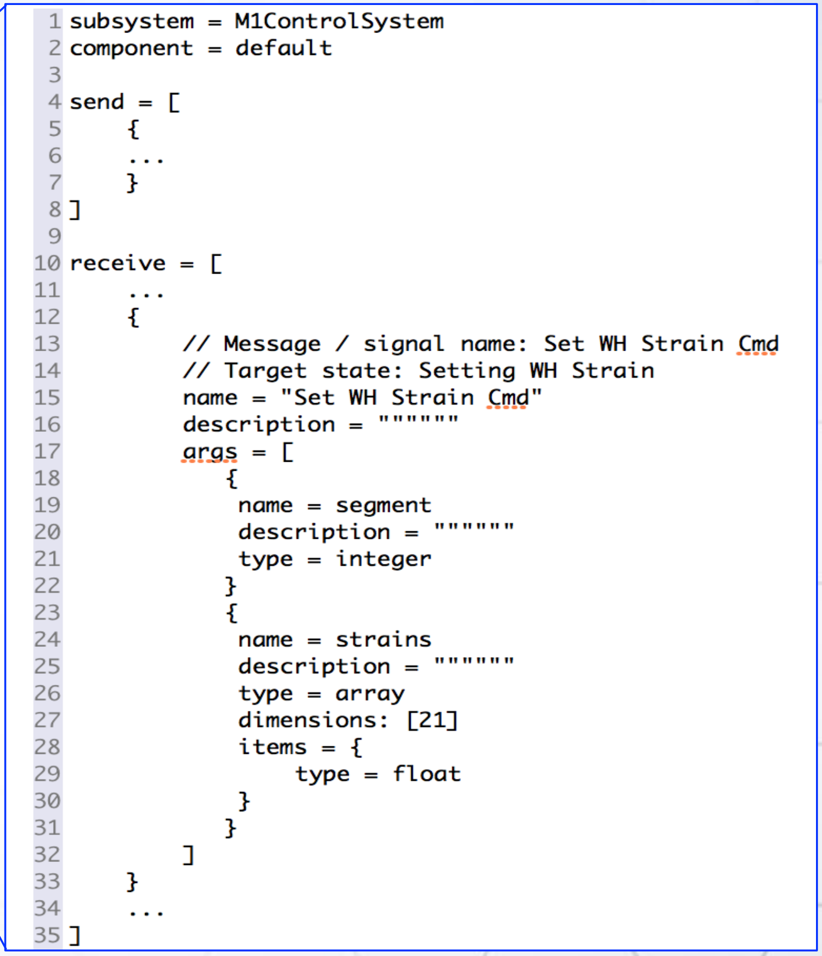
Extract Names

Extract Parameters

Verifying Conformance to Specified Interfaces



Comparison was performed manually – however, there is potential for automation



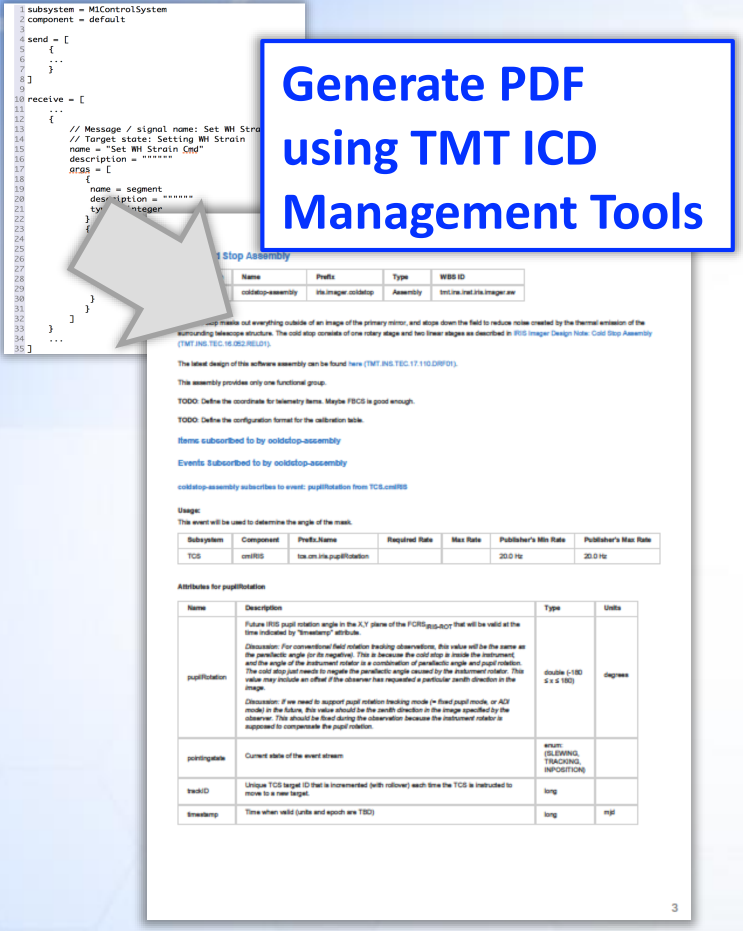
```
subsystem = M1ControlSystem
component = default

send = [
    {
    ...
    }
]

receive = [
    ...
    {
        // Message / signal name: Set WH Strain Cmd
        // Target state: Setting WH Strain
        name = "Set WH Strain Cmd"
        description = """"""
        args = [
           {
            name = segment
            description = """"""
            type = integer
           }
           {
            name = strains
            description = """"""
            type = array
            dimensions: [21]
            items = {
                type = float
            }
           }
        ]
    }
    ...
]
```

Generated ICD/API definition (in HOCON, TMT ICD Database Schema)

Generating Interface Control Documents



```
subsystem = M1ControlSystem
component = default

send = [
    {
    ...
    }
]

receive = [
    ...
    {
        // Message / signal name: Set WH Stra
        // Target state: Setting WH Strain
        name = "Set WH Strain Cmd"
        description = """"""
        args = [
            {
             name = segment
             description = """"""
             type = integer
            }
            {



             }
            }
        ]
    }
    ...
]
```

Generate PDF using TMT ICD Management Tools

Stop Assembly

Name	Prefix	Type	WBS ID
coldstop-assembly	iris.imager.coldstop	Assembly	tmt.ins.inst.iris.imager.sw

… masks out everything outside of an image of the primary mirror, and stops down the field to reduce noise created by the thermal emission of the surrounding telescope structure. The cold stop consists of one rotary stage and two linear stages as described in IRIS Imager Design Note: Cold Stop Assembly (TMT.INS.TEC.16.052.REL01).

The latest design of this software assembly can be found here (TMT.INS.TEC.17.110.DRF01).

This assembly provides only one functional group.

TODO: Define the coordinate for telemetry items. Maybe FBCS is good enough.

TODO: Define the configuration format for the calibration table.

Items subscribed to by coldstop-assembly

Events Subscribed to by coldstop-assembly

coldstop-assembly subscribes to event: pupilRotation from TCS.cmIRIS

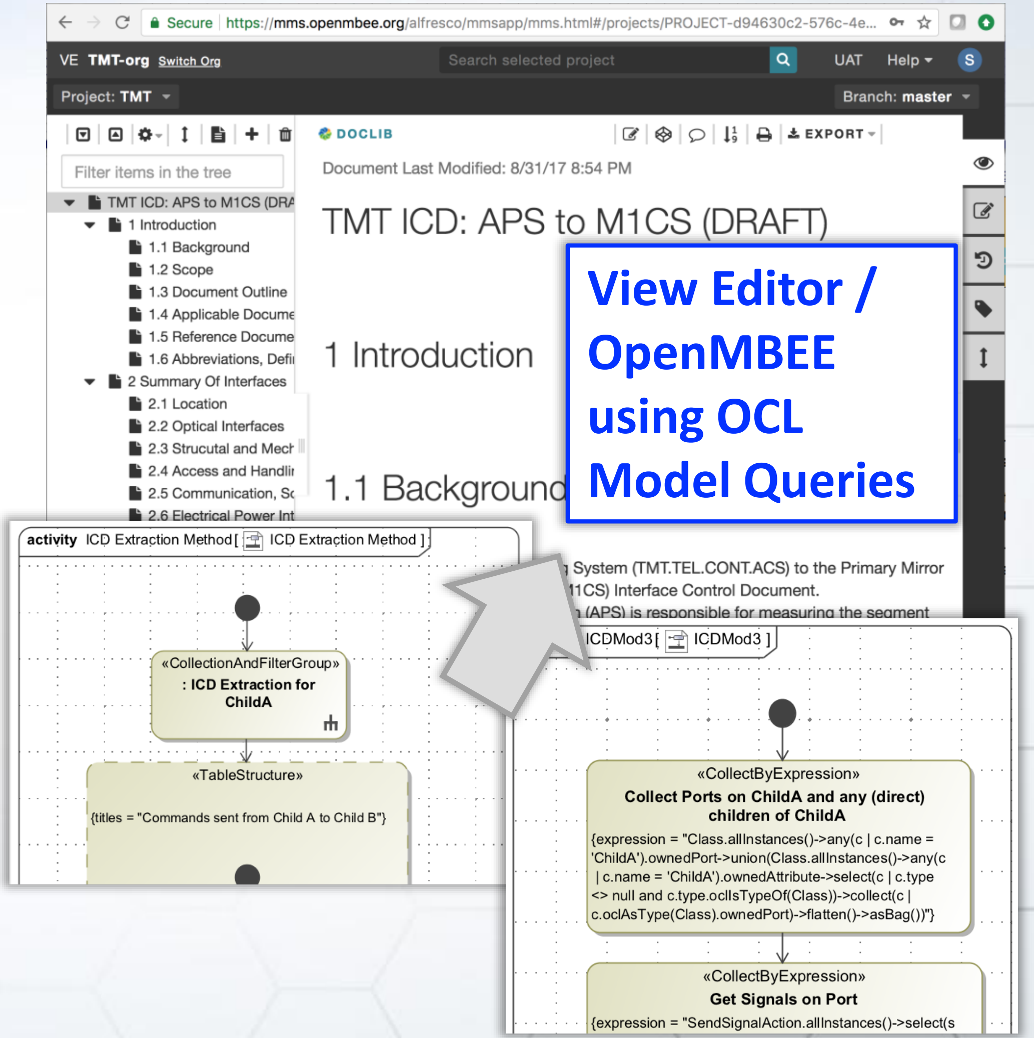
Usage:
This event will be used to determine the angle of the mask.

Subsystem	Component	Prefix.Name	Required Rate	Max Rate	Publisher's Min Rate	Publisher's Max Rate
TCS	cmIRIS	tcs.cm.iris.pupilRotation			20.0 Hz	20.0 Hz

Attributes for pupilRotation

Name	Description	Type	Units
pupilRotation	Future IRIS pupil rotation angle in the X,Y plane of the $FCRS_{IRIS-ROT}$ that will be valid at the time indicated by "timestamp" attribute. *Discussion: For conventional field rotation tracking observations, this value will be the same as the parallactic angle (or its negative). This is because the cold stop is inside the instrument, and the angle of the instrument rotator is a combination of parallactic angle and pupil rotation. The cold stop just needs to negate the parallactic angle caused by the insturment rotator. This value may include an offset if the observer has requested a particular zenith direction in the image.* *Discussion: If we need to support pupil rotation tracking mode (= fixed pupil mode, or ADI mode) in the future, this value should be the zenith direction in the image specified by the observer. This should be fixed during the observation because the instrument rotator is supposed to compensate the pupil rotation.*	double (-180 ≤ x ≤ 180)	degrees
pointingstate	Current state of the event stream	enum: (SLEWING, TRACKING, INPOSITION)	
trackID	Unique TCS target ID that is incremented (with rollover) each time the TCS is instructed to move to a new target.	long	
timestamp	Time when valid (units and epoch are TBD)	long	mjd

3



- TMT ICD: APS to M1CS (DRA
  - 1 Introduction
    - 1.1 Background
    - 1.2 Scope
    - 1.3 Document Outline
    - 1.4 Applicable Docume
    - 1.5 Reference Docume
    - 1.6 Abbreviations, Defi
  - 2 Summary Of Interfaces
    - 2.1 Location
    - 2.2 Optical Interfaces
    - 2.3 Strucutal and Mech
    - 2.4 Access and Handlir
    - 2.5 Communication, Sc
    - 2.6 Electrical Power Int

Document Last Modified: 8/31/17 8:54 PM

TMT ICD: APS to M1CS (DRAFT)

1 Introduction

1.1 Background

… System (TMT.TEL.CONT.ACS) to the Primary Mirror … 1CS) Interface Control Document.
… (APS) is responsible for measuring the segment

View Editor / OpenMBEE using OCL Model Queries

activity ICD Extraction Method [ICD Extraction Method]

«CollectionAndFilterGroup»
: ICD Extraction for ChildA

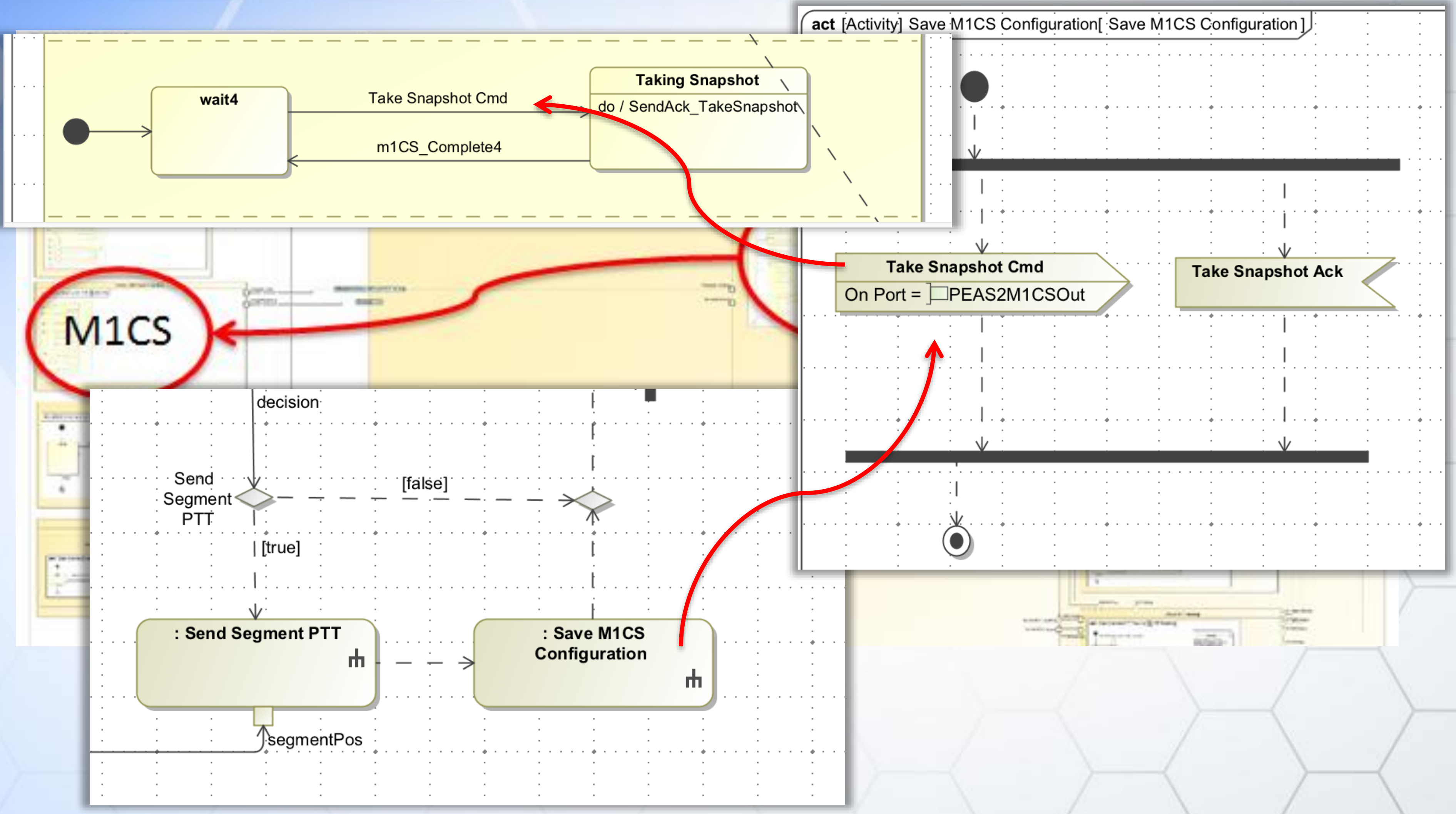
«TableStructure»
{titles = "Commands sent from Child A to Child B"}

ICDMod3 [ICDMod3]

«CollectByExpression»
Collect Ports on ChildA and any (direct) children of ChildA
{expression = "Class.allInstances()->any(c | c.name = 'ChildA').ownedPort->union(Class.allInstances()->any(c | c.name = 'ChildA').ownedAttribute->select(c | c.type <> null and c.type.oclIsTypeOf(Class))->collect(c | c.oclAsType(Class).ownedPort)->flatten()->asBag())"}

«CollectByExpression»
Get Signals on Port
{expression = "SendSignalAction.allInstances()->select(s

Application to APS-M1CS



Application to APS-M1CS: Specified vs. Extracted Commands

APS-M1CS ICD	SysML Model
setCalibCoeff	
getCalibCoeff	
offsetSegment	Move Segment PTT Cmd
saveCalibCoeff	
offloadSensorOffsets	offloadSensorOffsets Cmd
saveSensorReadings	Take Snapshot Cmd
genCalibCoeff	
calibrateWarpingHarness	Calibrate Warping Harness Cmd
readWHStrain	
setWHStrain	Set WH Strain Cmd
offsetWHStrain	
setWHPosition	Move Segment WH Cmd
offsetWHPosition	
	Turn WH On Cmd
	Turn WH Off Cmd

Unused commands: specified in ICD, not used in SysML model

Inconsistent naming schemes

Unspecified commands

Required human interpretation to find correspondences!

Application to APS-M1CS: Specified vs. Extracted Events

APS-M1CS ICD	SysML Model
m1cs.health	
m1cs.alarm	
m1cs.status	
m1cs.actuatorPositions	
m1cs.sensorHeights	m1cs.sensorHeights Cmd
m1cs.sensorGaps	
m1cs.pistonTipTilt	Get Segment WH Pos Cmd
m1cs.servoErrors	
m1cs.pistonTipTiltTarget	
m1cs.outerLoopCtrlCmds	
m1cs.segmentStatus	Get installed_Segment_Query
m1cs.warpingHarnessStrain	Get Segment WH Pos Ack
m1cs.warpingHarnessStatus	
m1cs.purgeSystemStatus	
m1cs.ctrlNetworkStatus	

Large number of unused events – incomplete model?

Inconsistent naming scheme

Due to limited language vocabulary, could not differentiate between commands & pub/sub events!

Application to APS-M1CS: Parameters

Parameter names do not always match

Some parameters missing

ICD Command / Event	ICD Parameters		Extracted Parameters	
	Name	*Type*	*Name*	*Type*
offsetSegment	actuatorOffset	[492x3]		
saveSensorReadings	type	[ALIGNED, DIAGNOSTIC]		
	metadata	string		
offloadSensorOffsets	segmentLocation	integer	segmentLocation	integer
calibrateWarpingHarness	motor number	integer	motorID	integer
	segment	integer	segment	integer
setWHStrain	segment	integer	segment	integer
	strains	float [21]	strains	float [21]
setWHPosition	segment	integer	p	double
	position	integer [21]		
m1cs.sensorHeights	heights	float [2272]		
m1cs.warpingHarnessStrain	strain	float [492x21]		
m1cs.pistonTipTilt	pistonTipTilt	float [3]		
m1cs.warpingHarnessStrain	strain	float [492x21]	p	double

Mismatched parameters

Summary & Conclusions

- Possible to extract *core* software interface information: some information could not be extracted (timing, frequencies, etc.)
- Detected many discrepancies between specified interfaces, and actual interface derived from specified behavior
  - Discovered use of outdated versions of APIs / interfaces, and use of non-existent API calls
  - Some differences may be result of wrong assumptions on what component performs function (e.g., storing of actuator position data)
  - Discrepancies have impact on timing and other resources, and affects whether or not requirements are satisfied
- Need better interface mgmt. in MBSE with SysML applications
  - Semantic variation points in UML / SysML
  - Native SysML vocabulary not sufficient to differentiate between pub / sub events and command invocation → **need vocabulary extensions**

Acknowledgments

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and NoMagic.

The TMT Project gratefully acknowledges the support of the TMT collaborating institutions. They are the Association of Canadian Universities for Research in Astronomy (ACURA), the California Institute of Technology, the University of California, the National Astronomical Observatory of Japan, the National Astronomical Observatories of China and their consortium partners, and the Department of Science and Technology of India and their supported institutes. This work was supported as well by the Gordon and Betty Moore Foundation, the Canada Foundation for Innovation, the Ontario Ministry of Research and Innovation, the National Research Council of Canada, the Natural Sciences and Engineering Research Council of Canada, the British Columbia Knowledge Development Fund, the Association of Universities for Research in Astronomy (AURA) and the U.S. National Science Foundation.

References

- Open Source TMT model: https://github.com/Open-MBEE/TMT-SysML-Model
- A Practical Guide to SysML, 3rd Edition, Chapter 17 by Friedenthal, Moore, and Steiner
- OMG Unified Modeling Language (OMG UML) 2.5.1 Specification: https://www.omg.org/spec/UML/2.5.1/
- OMG Systems Modeling Language (OMG SysML) 1.5 Specification: https://www.omg.org/spec/SysML/1.5/
- Karban, R., Dekens, F. G., Herzig, S., Elaasar, M., and Jankevicius, N., *“Creating system engineering products with executable models in a model based engineering environment,”* Modeling, Systems Engineering, and Project Management for Astronomy VI, SPIE, Edinburgh, UK (2016).
- Herzig, S., Karban, R., Trancho, G., Dekens, F., Jankevicius, N., Troy, M., *“Analyzing the Operational Behavior of the Alignment and Phasing System of the Thirty Meter Telescope,”* Adaptive Optics for Extremely Large Telescopes (AO4ELT), Tenerife, Spain (2017).